REMARKS/ARGUMENTS

Prior to entry of this amendment, claims 1-21 were pending in this application. Claims 1, 5, 9, 10, and 13 have been amended, no claims have been added, and no claims have been canceled herein. Therefore, claims 1-21 remain pending. Applicant respectfully requests reconsideration of these claims for at least the reasons presented below.

35 U.S.C. § 103 Rejection, Karkare in view of Gove

Claims 1-21 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,266,810 to Karkare et al. (hereinafter "Karkare") in view of U.S. Patent Pub. No. 2004/0006760 of Gove et al. (hereinafter "Gove"). The Applicants respectfully submit that the Office Action does not establish a *prima facie* case of obviousness in rejecting these claims, as amended. Therefore, the Applicants request reconsideration and withdrawal of the rejection.

In order to establish a *prima facie* case of obviousness, all claimed limitations must first be taught or suggested by the prior art. *See, e.g.*, *DyStar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co.*, 464 F.3d 1356, 1360 (Fed. Cir. 2006). The Office Action must then provide an explicit analysis supporting the rejection. *See KSR Int'l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1741 (2007) ("a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art"). While the Office Action can use one of several exemplary rationales from the MPEP to support an obviousness rejection under *KSR*, all the rationales still require the Office Action to demonstrate that all the claim elements are shown in the prior art. *See* MPEP §2143. As will be discussed below, the references cited by the Office Action do not teach or suggest each claimed limitation. For example, none of the references, alone or in combination, teach or suggest determining an application component of an application to be monitored in the IDE, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment as recited in independent claims 1 and 9 or a component repository configured to maintain a list of available application components that can be invoked by an integrated development environment ("IDE") runtime environment to monitor an application, the application comprising operational logic and business logic and monitoring operation of the application component to determine a plurality of metrics associated with the application component wherein the application component is selected from the operational logic of the application as recited in independent claim 10.

Karkare is directed to "the runtime profiling of a software application and the platform-dependent code component(s) used by the software application." (Col. 1, lines 9-11) More specifically, Karkare describes "a profiling system for runtime environments that include a software application written in a platform-independent programming language, a non-application-code component invoked by the software application and a profiling tool for generating a runtime metric." (Col. 2, lines 36-41) However, Karkare does not teach or suggest an application comprising operational logic and business logic. Rather, Karkare describes "application-code components" (see for example col. 4, line 7 – col. 5, line 35) and "non-application-code components" (see for example col. 5, line 36 – col. 6, line 42) However, Karkare does not describe the application-code components comprising business logic and operational logic. Furthermore, any reading of Karkare that would equate the application-code logic to business logic and the non-application-code logic to operational logic would not be reasonable since, as the name "non-application-code components" implies and as explicitly stated in Karkare, "non-application-code includes any code component that does not reside within the application code component" (col. 5, lines 43-45), i.e., it is separate from the application-code component. Thus, Karkare does not describe an application comprising operational logic and business logic and further monitoring a component of such an application selected from the operational logic of the application. Therefore, the Applicants respectfully contend that Karkare does not teach or suggest determining an application component of an application to be monitored, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of an IDE runtime environment.

Gove is directed to "using an integrated development environment (IDE) and more particularly to using profiles in an integrated development environment to improve application performance." (paragraph 2) Under Gove, a developer codes a program, the IDE executes the code and collects performance data, and the developer reviews the performance data and debugs the program. (See paragraph 20) However, Gove does not characterize or distinguish contents of the program being executed and debugged. That is, Gove does not describe the program comprising business logic and operational logic. Therefore, the Applicants respectfully contend that Gove does not teach or suggest, alone or in combination with Karkare, determining an application component of an application to be monitored, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of an IDE runtime environment.

Claim 1, upon which claims 2-8 depend, recites in part "determining an application component of an application to be monitored in the IDE, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment; [and] monitoring the application component in the IDE runtime environment to determine a plurality of metrics associated with the application component." Neither Karkare nor Gove, alone or in combination, teaches or suggests determining an application component of an application to be monitored in the IDE, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment and monitoring the application component in the IDE runtime environment. For at least these reasons, the Applicants respectfully request reconsideration and withdrawal of the rejection.

Claim 9 recites in part "determine an application component of an application to be monitored in the IDE, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment; [and] monitor the application component in the IDE runtime environment to determine a plurality of metrics associated with the application component." Neither Karkare nor Gove, alone or in combination, teaches or suggests determining an application component of an application to be monitored in the IDE, the application comprising operational logic and business logic, the application component being selected from the operational logic of the application and having associated information in a component repository of the IDE runtime environment, wherein the component repository is configured to provide a list of components that are available to be invoked by the IDE runtime environment and monitoring the application component in an IDE runtime environment. For at least these reasons, the Applicants respectfully request reconsideration and withdrawal of the rejection.

Claim 10, upon which claims 11-21 depend, recites in part "a component repository configured to maintain a list of available application components that can be invoked by an integrated development environment ("IDE") runtime environment to monitor an application, the application comprising operational logic and business logic; [and] an IDE runtime environment configured to open an application component and monitor operation of the application component to determine a plurality of metrics associated with the application component wherein the application component is selected from the operational logic of the application." Neither Karkare nor Gove, alone or in combination, teaches or suggests a component repository configured to maintain a list of available application components that can be invoked by an integrated development environment ("IDE") runtime environment to monitor an application, the application comprising operational logic and business logic and monitoring operation of the application component to determine a plurality of metrics associated with the application component wherein the application component is selected from the operational logic of the application. For at least these reasons, the Applicants respectfully request reconsideration and withdrawal of the rejection.

The various dependent claims are thought to be allowable for additional reasons. For example, claim 5 recites in part "providing a policy manager in the IDE to allow the user to select the application component and specify an operational concern for the application component the operational concern related to the business logic of the application; communicating the specified operational concern to a policy agent in the IDE runtime environment; and enforcing the operational concern with the policy agent during operation of the application component." None or the references, alone or in combination, teach or suggest a policy manager in the IDE to allow the user to select an application component and specify an operational concern, i.e., policy, for the application component where the operational concern (policy) is related to the business logic of the application. Furthermore, none of the references teach or suggest, alone or in combination, communicating the specified operational concern to a policy agent in the IDE runtime environment and enforcing the operational concern with the policy agent during operation of the application component. Claim 13 includes similar recitations and is also thought to be allowable for at least these same additional reasons.

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance and an action to that end is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 303-571-4000.

Respectfully submitted,

/William J. Daley/
William J. Daley
Reg. No. 52,471

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 303-571-4000
Fax: 415-576-0300

WJD:jep
62062601 v1